## AMENDMENTS TO THE SPECIFICATION

*Please replace paragraph [0020] on pages 7-8 of the application as filed with the following amended paragraph.*

The present invention is directed to a network device that receives data and ~~process~~ processes that data and may forward that data onto a destination based on attributes of that data. A general schematic of one such network device 101, according to one embodiment of the present invention, is illustrated in Fig. 1. This device integrates eight 10/100BASE-TX transceivers 110-117, one general use (10/100BASE-TX/FX) MII, nine full-duplex capable Media Access Controllers (MACs) 120-127 and 109, a Serial Management Port 105, high-performance integrated packet buffer memory 135, an address resolution engine, a non-blocking switch controller, and a set of Management Information Base (MIB) statistics registers 131.

*Please replace paragraph [0023] on pages 8-9 of the application as filed with the following amended paragraph.*

An integrated address management engine provides address learning and recognition functions at maximum frame rates. The address table provides capacity for up to 4K unicast and multicast addresses. Addresses are added to the table after receiving an error-free packet. Broadcast and multicast frames are forwarded to all ports within the VLAN domain except the port where it was received. The network devices may be cascaded to 36 ports through 3.2 Gbps expansion [[port]] ports or stacked to 48 ports with 200Mbps turbo MII [[port]] ports.

*Please replace paragraph [0024] on page 9 of the application as filed with the following amended paragraph.*

Address learning is one of the key features in the design of network devices. When a frame is received at a network device, the receiving port [[with]] will use the destination address to determine which destination port the frame should be forwarded to. If the destination address lookup produces a match, the receiving port will be able to know which port or ports this frame is destined for. The receiving port will then relay the frame only to the destination port or ports. If the destination address ~~look up~~ lookup fails, the receiving port cannot find to which port or ports the frame is destined. The receiving port will then relay the frame to all ports. (This is called flooding the frame.)

*Please replace paragraph [0026] on page 10 of the application as filed with the following amended paragraph.*

An example of the above-described process is also illustrated in Fig. 2. When a frame is received by the network device [[201]] (block 201), the destination address contained within the frame is compared with entries in the ARL table for the network device [[202]] (block 202) to determine if one of the entries can be matched against the destination address [[203]] (block 203). If a match is found, the frame is relayed to the destination port of the network device corresponding to the destination address obtained from the appropriate entry in the ARL table [[205]] (block 205). If no match is determined, then the frame is relayed or flooded to all ports of the network device [[204]] (block 204). Then, a lookup in the ARL table is performed for the source address [[206]] (block 206) to determine whether the source address has already been learned [[207]] (block 207). If the source address has been previously learned, then the hit bit~~for that entry~~ is updated for that entry in the ARL table [[208]] (block 208). If the source address has not already been learned, then an entry is made in the ARL table for the source address [[209]] (block 209). Additionally, learning messages are sent out so that other network devices connected can also learn the source address [[210]] (block 210).

*Please replace paragraph [0027] on pages 10-11 of the application as filed with the following amended paragraph.*

One example of the above-described~~operation~~ operations is provided herein. At a first time, port 1 receives a frame with a source address equal to MAC_1. If the source address lookup of the port 1 fails, port 1 will update the ARL table~~with~~ indicating that MAC_1 is located at port 1. Additionally, port 1 will send a frame to other linked network devices so that so that the devices learn that MAC_1 is destined for port 1. Subsequently, port 2 receives a frame with a destination address equal to MAC_1, port 2 will perform a destination address lookup. Port 2 will then determine that MAC_1 is located at port 1, as that port was already learned by the ARL table at the earlier time. Thus, port 2 will forward the frame to port 1.

*Please replace paragraph [0030] on pages 11-12 of the application as filed with the following amended paragraph.*

When device 0 receives frame2to1, because it has a MAC_1 entry in its ARL table already, device 0 will relay frame2tol to port 1 only. Since there is no frame2tol going to device 1, device 1 will not have the opportunity to learn MAC_2. Therefore, using data~~ frame~~ frames alone cannot guarantee [[a]] synchronization of ARL table entries in a multi-device configuration.

*Please replace paragraph [0037] on page 14 of the application as filed with the following amended paragraph.*

In part, this can be accomplished by setting a learned-all-devices tag in the ARL table. This tag is used to tell the receiving port logic that the corresponding MAC address has been learned by all connected network devices, and thus, that no learning message need be sent regarding the MAC address. When there are a lot of new MAC address that need to be learned, some of the learning messages may be lost and~~ other~~ others may be learned. Only the MAC addresses that have not yet been learned by all devices are re-sent by the corresponding receiving ports.

*Please replace paragraph [0038] on pages 14 of the application as filed with the following amended paragraph.*

One example of the above-described process is also illustrated in Fig. 5. When a frame is received by the network device [[501]] (block 501), the destination address contained within the frame is compared with entries in the ARL table for the network device [[502]] (block 502) to determine if one of the entries can be matched against the destination address [[503]] (block 503). If a match is found, the frame is relayed to the destination port of the network device corresponding to the destination address obtained from the appropriate entry in the ARL table [[505]] (block 505). If no match is determined, then the frame is relayed or flooded to all ports of the network device [[504]] (block 504). Then, a lookup in the ARL table is performed for the source address [[506]] (block 506) to determine whether the source address has already been learned [[507]] (block 507). If the source address has been previously learned, then the hit bit~~ for~~

~~that entry~~ is updated for that entry in the ARL table [[508]] (block 508). If the source address has not already been learned, then an entry is made in the ARL table for the source address [[509]] (block 509).

*Please replace paragraph [0039] on pages 14-15 of the application as filed with the following amended paragraph.*

As compared to Fig. 2, the flowchart in Fig. 5 has additional logic such that a determination is made whether all connected network devices have learned the given address [[520]] (block 520). If all of the connected devices have learned the address, then the flow continues to await the next frame. If all connected network devices have not learned the address, then the learning message continues to be sent [[510]] (block 510), where learning messages are also sent out after step [[509]] (block 509) so that other network devices connected can also learn the source address.